

DFW



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application No. : 10/739,222 Confirmation No.: 8472
Applicant : Kentaro YOSHIMURA, et al.
Filed : December 19, 2003
TC/A.U. : 2122
Examiner : To Be Assigned
Docket No. : 056207.53098US
Customer No. : 23911
Title : Embedded Controllers and Development Tool for Embedded
Controllers

CLAIM OF PRIORITY UNDER 35 U.S.C. § 119

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

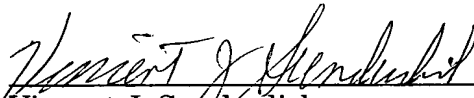
Sir:

The benefit of the filing date of prior foreign application No. 2002-369167, filed in Japan on December 20, 2002, is hereby requested and the right of priority under 35 U.S.C. § 119 is hereby claimed.

In support of this claim, filed herewith is a certified copy of the original foreign application.

Respectfully submitted,

September 10, 2004



Vincent J. Sunderdick
Registration No. 29,004

CROWELL & MORING LLP
Intellectual Property Group
P.O. Box 14300
Washington, DC 20044-4300
Telephone No.: (202) 624-2500
Facsimile No.: (202) 628-8844
VJS:ms #337135

日本国特許庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: 2002年12月20日

出願番号
Application Number: 特願2002-369167

[ST. 10/C]: [JP 2002-369167]

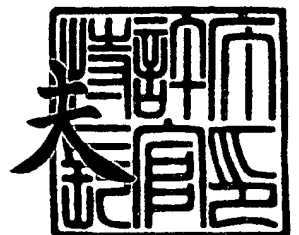
願人
Applicant(s): 株式会社日立製作所

CERTIFIED COPY OF
PRIORITY DOCUMENT

2003年 9月11日

特許庁長官
Commissioner,
Japan Patent Office

今井康夫



【書類名】 特許願

【整理番号】 1102020891

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 9/44

【発明の名称】 組込みコントローラ及び組込みコントローラ開発ツール

【請求項の数】 17

【発明者】

【住所又は居所】 茨城県日立市大みか町七丁目 1 番 1 号
株式会社 日立製作所 日立研究所内

【氏名】 吉村 健太郎

【発明者】

【住所又は居所】 茨城県日立市大みか町七丁目 1 番 1 号
株式会社 日立製作所 日立研究所内

【氏名】 宮崎 泰三

【発明者】

【住所又は居所】 茨城県ひたちなか市大字高場 2 5 2 0 番地
株式会社 日立製作所 自動車機器グループ内

【氏名】 横山 孝典

【特許出願人】

【識別番号】 000005108

【氏名又は名称】 株式会社 日立製作所

【代理人】

【識別番号】 100075096

【弁理士】

【氏名又は名称】 作田 康夫

【電話番号】 03-3212-1111

【手数料の表示】

【予納台帳番号】 013088

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 組込みコントローラ及び組込みコントローラ開発ツール

【特許請求の範囲】

【請求項 1】

センサによって計測した入力信号または他の組込みコントローラから受信した入力信号の少なくともいずれかの入力信号をデータ化する外部入力処理手段と、複数の制御演算手段を有すると共に前記データ化された入力信号または当該および他の制御演算手段の少なくともいずれかにより求められた制御データに基づいて当該演算手段が制御データを求め、前記当該演算手段により求められた制御データに基づいてアクチュエータを駆動する制御信号を生成する外部出力処理手段と、前記外部入力処理手段と前記外部出力処理手段と前記制御演算手段の実行順序を管理する制御基本処理手段と、前記制御基本手段からの要求により前記制御演算手段への演算実行命令と前記当該制御演算手段と他の制御演算手段との間の制御データの受け渡しを管理するインタフェース手段を制御演算手段ごとに設けることを特徴とする組込みコントローラ。

【請求項 2】

請求項 1 記載の組込みコントローラにおいて、

前記インタフェース手段は、前記インタフェース手段に対応する前記制御演算手段によって計算された前記制御データを保存する事を特徴とする組込みコントローラ。

【請求項 3】

請求項 2 記載の組込みコントローラにおいて、

前記インタフェース手段は、前記インタフェース手段が保存した前記制御演算手段の前記制御データを他のインタフェース手段が参照することを特徴とする組込みコントローラ。

【請求項 4】

請求項 3 記載の組込みコントローラにおいて、

前記インタフェース手段は、前記インタフェース手段に対応する前記制御演算手段が計算を行う際に用いる前記データを、他のインタフェース手段が保存する

前記制御データもしくは当該インタフェース手段に保存された前記制御データを参照して収集することを特徴とする組込みコントローラ。

【請求項 5】

請求項 4 記載の組込みコントローラにおいて、

前記インタフェース手段は、前記制御演算手段が計算を行う際に、前記インタフェース手段が収集した前記制御演算手段が演算に用いる前記データを参照できる状態にして、前記制御演算手段によって計算されるデータが保存される前記制御データを指定して、前記制御演算手段の演算処理実行を命令することを特徴とする組込みコントローラ。

【請求項 6】

請求項 1, 2, 3, 4, 5 のいずれかに記載の組込みコントローラにおいて、

前記インタフェース手段は、前記制御演算手段の計算結果である制御データを保存すると共に、前記制御演算手段ごとの制御データは 1 つのみ保存することを特徴とする組込みコントローラ。

【請求項 7】

請求項 1, 2, 3, 4, 5, 6 のいずれかに記載の組込みコントローラにおいて、

前記制御演算手段は演算において参照するデータと特定の演算手順と演算結果を出力する前記制御データとを記述した制御モデルに基づいて、プログラミング言語のソースコードが自動生成されたものであることを特徴とする組込みコントローラ。

【請求項 8】

特定の計算手順に基づく制御演算手段によって計算された制御データの出力および保存と、前記制御演算手段を含んだ複数個の前記制御演算手段への前記制御データの提供とを行うインタフェース手段を生成する組込みコントローラ開発ツールであって、

前記制御演算手段をプログラムのソースコードの中で関数として記述し、前記制御演算手段が計算に用いる参照データは関数の引数であると共に、前記制御演算手段によって計算された前記制御データは、前記関数の戻り値またはアドレス

を指定した関数の引数とし、

前記制御演算手段のソースコードから特定の情報を抽出する制御演算手段解析手段と、前記制御演算手段解析手段の解析結果に基づいて前記制御演算手段ごとに対応する前記インタフェース手段をプログラムのソースコードとして生成するインタフェース手段生成手段を有することを特徴とする組込みコントローラ開発ツール。

【請求項 9】

請求項 8 記載の組込みコントローラ開発ツールにおいて、

インタフェース手段生成手段は、制御演算手段解析手段により抽出した制御演算手段の計算結果である前記制御データの制御データ名と計算結果である前記制御データの型情報に基づいて、インタフェース手段生成手段が前記制御データの保存領域を組込みコントローラ上に割り当てるための制御データ宣言をインタフェース手段のソースコードに設ける事を特徴とする組込みコントローラ開発ツール。

【請求項 10】

請求項 8 記載の組込みコントローラ開発ツールにおいて、

インタフェース手段生成手段は、制御演算手段解析手段により抽出した計算に用いる前記参照データの数と計算に用いる前記参照データの参照データ名とに基づいて、インタフェース手段生成手段がインタフェース手段のソースコードに外部入力処理手段や他のインタフェース手段から計算に用いるデータを読み出す機能を設ける事を特徴とする組込みコントローラ開発ツール。

【請求項 11】

請求項 8 記載の組込みコントローラ開発ツールにおいて、

インタフェース手段生成手段は、制御演算手段解析手段により抽出した制御演算手段の計算結果である前記制御データの制御データ名に基づいて、インタフェース手段生成手段がインタフェース手段のソースコードに前記制御基本処理手段などが制御演算手段の計算結果である前記制御データの更新を要求するための関数またはマクロ命令を設けることを特徴とする組込みコントローラ開発ツール。

【請求項 12】

請求項 8 記載の組込みコントローラ開発ツールにおいて、

インタフェース手段生成手段は、制御演算手段解析手段により抽出した制御演算手段の計算結果である前記制御データの前記制御データ名に基づいて、インタフェース手段生成手段がインタフェース手段のソースコードの中に他のインタフェース手段や外部出力処理手段が制御演算手段の計算結果である前記制御データを参照するための関数またはマクロ命令を設けることを特徴とする組込みコントローラ開発ツール。

【請求項 13】

請求項 8, 9, 10, 11, 12 のいずれかに記載の組込みコントローラにおいて、

前記制御演算手段は演算において参照する前記参照データと特定の演算手順と演算結果を出力する前記制御データとを記述した制御モデルに基づいて、プログラミング言語のソースコードが自動生成されたものであることを特徴とする組込みコントローラ開発ツール。

【請求項 14】

請求項 8, 9, 10, 11, 12, 13 のいずれかに記載の組込みコントローラ開発ツールを用いて開発された組込みコントローラ。

【請求項 15】

センサによって計測する運転者の指令とエンジン状態または他の車両制御コントローラから受信する外部からの入力信号に基づいて、内燃機関または変速機を制御する制御信号を出力する車両制御コントローラにおいて、

センサによって計測または他の組込みコントローラから受信する外部からの入力信号をデータ化する外部入力処理手段と、入力信号のデータや他の制御演算手段の計算結果から特定の計算手順に基づいてデータの計算を行う複数の制御演算手段と、制御演算手段により計算されたデータに基づいて内燃機関または変速機を制御する制御信号を出力する外部出力処理手段と、前記外部入力処理手段と前記外部出力処理手段と前記制御演算手段の実行順序を管理する制御基本処理手段と、前記制御演算手段ごとに対応させて前記制御基本手段から前記制御演算手

段への実行命令と前記制御演算手段へのデータの受け渡しを管理するインタフェース手段とを設けることを特徴とする車両制御コントローラ。

【請求項 16】

請求項 15 に記載の車両制御コントローラにおいて、
前記制御演算手段は演算において参照するデータと特定の演算手順と演算結果を出力するデータとを記述した車両制御モデルに基づいて、プログラミング言語のソースコードが自動生成されたものであることを特徴とする車両制御コントローラ。

【請求項 17】

請求項 8, 9, 10, 11, 12, 13 のいずれかに記載の組込みコントローラ開発ツールは、前記車両制御コントローラを開発することを特徴とするコントローラ開発ツール。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、組込みコントローラ及び組込みコントローラ開発ツールに関する。

【0002】

【従来の技術】

従来より組込みソフトウェア、組込みコントローラおよび組込みソフトウェア開発ツールとして、プログラム作成者からの入力情報に従って、基本プログラムのインタフェースプログラムを自動的に生成する方法がある（例えば、特許文献 1 参照）。

【0003】

また従来、組込みソフトウェアを作成するために制御系システム設計支援ソフトウェア、引用例によれば MATLAB および Simulink を用いて、ソースコードを自動生成する方法がある（例えば、非特許文献 1 参照）。

【0004】

【特許文献 1】

特開 2002-229791 号公報（第 4 頁，図 1（b））

【非特許文献 1】

サイバネットシステム株式会社刊、「MATLAB Expo2002 モデルベース制御系設計カンファレンス資料」、103頁～126頁

【0005】**【発明が解決しようとする課題】**

例えば車両用電子制御装置においては、組込みコントローラ用ソフトウェアの開発効率を向上させるための組込みコントローラ開発ツールおよび開発プロセスが研究されている。開発効率向上のためには、組込みコントローラ用ソフトウェアのソースコード自動生成化および再利用性向上が要求されている。

【0006】

従来の組込みコントローラでは、インタフェース手段が基本管理手段の一部である（例えば、特許文献 1 参照）。そのため、複数ある制御演算手段の一部が変更された場合でも、インタフェース手段全体を変更する必要がある、本来変更する必要のないインタフェース手段の部分を作り直すことになってしまうという課題がある。本発明は、上記課題を解決し、組込みコントローラの再利用性を向上させることを目的とする。

【0007】

従来、MATLAB および Simulink を用いて制御演算手段のソースコードを自動生成する方法がある（例えば、非特許文献 1 参照）。上記の方法で制御演算手段を生成したとしても、自動生成された制御演算手段と制御演算手段とのデータの受け渡しや組込みコントローラ全体で用いる変数の管理といった統合化作業の際に、手作業が必要であった。人間によりインタフェース手段のソースコードを記述する場合、ソースコードの記述に時間がかかるという課題がある。本発明は、上記課題を解決し組込みコントローラの生産性を向上させることを目的とする。

【0008】**【課題を解決するための手段】**

本発明では、組込みコントローラにおいて、制御演算手段毎に対応するインタフェース手段を設けることにより、上記目的を達成する。

【0009】

本発明では、組込みコントローラ開発ツールにおいて、前記制御演算手段がプログラミング言語のソースコードで関数として記述されて、センサ計測値や他の制御演算手段の計算結果などの計算に用いる変数は関数の引数で、計算結果である変数は関数の戻り値またはメモリ上のアドレスを指定した関数の引数であることと、前記インタフェース手段がプログラミング言語のソースコードで記述されることとを前提として、前記制御演算手段のソースコードから特定の情報を抽出する制御演算手段解析手段と、制御演算手段解析手段の解析結果に基づいて制御演算手段に対応する前記インタフェース手段のソースコードを生成するインタフェース手段生成手段を設けることにより、上記目的を達成する。

【0010】

【発明の実施の形態】

以下に本発明の実施の一形態を図面と共に説明する。なお、本実施例では組込みコントローラの一例として車両制御コントローラを実施例として扱う。

【0011】

まず、図1に本発明が実施されたアプリケーションソフトウェアの基本構造を示す。1A, ..., 1B, 1Cは制御変数の更新ロジックを記述した制御ソフト部品である。これは、本発明における制御演算手段に対応する。2A, ..., 2B, 2Cは制御ソフト部品に対応したインタフェースソフトである。これは、本発明におけるインタフェース手段に対応する。3は本アプリケーションで実行される基本ロジック、例えば変数の更新順序すなわち制御ソフト部品の実行順序を記述した制御基本ソフトウェアである。これは、本発明における制御基本処理手段に対応する。4は外部入力データであって、例えばハードウェアの持つセンサで計測したり他のアプリケーションおよびコントローラと通信することによって、前記外部情報を得る。これは、本発明における外部入力処理手段に対応する。5は外部への情報の出力であって、ハードウェアの持つアクチュエータを駆動したり他のアプリケーションおよびコントローラと通信することによって、前記外部出力を実行する。本発明における外部出力処理手段に対応する。6が組込みソフトウェアのタスク制御や割込み管理を行うオペレーションシステムをあらわしてい

る。本発明における基本管理手段に対応する。

【0012】

ソフト部品とインタフェースソフトはそれぞれ1Aと2A, 1Bと2Bといったように対応付けられている。そのため、例えばソフト部品1Cが変更された場合にはインタフェースソフト2Cのみを変更すればよく、インタフェースソフト2A, 2Bは変更せずに用いる事ができる。すなわち図1の構成とすることによって、インタフェースソフトを有する組込みソフトウェアの再利用性が向上するという効果がある。

【0013】

なお、図1ではアプリケーションの一つの階層においてのみ表現しているが、本発明は組込みソフトウェアのある一つの階層に限定したものではなく、あらゆる階層において実施する事ができる。たとえばソフト部品1A自身が図1に示した構成を持つこともできる。この構成は、いわゆる階層化された組込みソフトウェア構造において有効である。

【0014】

図2に、本発明を適用した車両を示す。この例では、車両7は、エンジン8と、エンジン8を制御する制御ユニット(ECU)10と、例えば、自動変速機(AT)9と、自動変速機9を制御する制御ユニット(ECU)20と、ドライバにより操作されるアクセルペダルAを有する。またここで、制御ユニット(ECU)20は、自動変速機に関するものの他、スロットル制御のための制御ユニット等であっても良く、エンジンを制御するユニット以外の制御ユニットを示したものである。

【0015】

図3に、車両用分散制御システムの基本構成を示す。

【0016】

本車両用分散制御システムは、

プロセッサ13と、メモリ12と、組込みソフトウェア11と、データ受信装置14と、データ送信装置15と、エアフローセンサ等の入力をA/D変換して入力するアナログ入力41と、クランク角度センサ等からのパルスを入力するデ

デジタル入力 42 と、周辺装置を駆動する為に電圧を出力するアナログ出力 51 と、周辺装置を駆動するためにパルスを出力するデジタル出力 52 とによって構成される制御ユニット (ECU) 10 と、

前記制御ユニット (ECU) 10 と同様の構成を有する制御ユニット (ECU) 20 とで構成されている。

【0017】

図 4 に、組込みソフトウェア 11 の構成の一例を示す。図 4 に示すように、組込みソフトウェア 11 はアプリケーションソフトウェア 111 と基本ソフトウェア 112 で構成されている。

【0018】

基本ソフトウェア 112 は、リアルタイムオペレーションシステム (RTOS) 1121 と基本入出力システム (BIOS) 1122 とで構成されている。RTOS 1121 は、時間周期タスクの起動や外部入力に同期したタスクの起動を行い、割込み禁止処理や他のタスクの呼び出しなどのサービスをアプリケーションソフトウェア 111 に提供する。BIOS 1122 は、アプリケーションソフトウェア 111 からの要求を受けて、外部入力データ 4 の読み出しや、外部出力データ 5 の出力を実行する。

【0019】

図 4 の構成にすることによって、組込みソフトウェアの中でハードウェアに依存する部分 (BIOS) と、割込み信号処理やタイマー処理などのリアルタイム制御機能と、変速機の変速制御やエンジンの点火制御など制御対象に依存するアプリケーション部分とを分離する事ができる。例えば CPU が変わった場合には BIOS のみを変更すればよく、アプリケーションは流用する事ができるという利点がある。

【0020】

図 5 に、アプリケーションソフトウェアの構成の一例を示す。アプリケーションソフトウェア 111 は例えば 10 [ms] 周期タスク 1111 のように一定周期で起動するタスクと、エンジン回転同期タスク 1112 のように外部信号に同期して起動するタスクと、バックグラウンドタスク 1113 のようにプロセッサ

が空き状態のときに実行されるタスクとで構成されている。

【0021】

それぞれのタスクはタスク処理基本ソフトウェアと制御アプリケーションとで構成されている。例えば10 [ms] 周期タスクは、10 [ms] タスク処理基本ソフトウェアと診断制御11112と燃料補正制御11113とトルクベース制御11114とその他の制御で構成されている。図中の「…」は、制御ロジックが上記の3種のみには限らないことを示している。

【0022】

図6に10 [ms] タスク処理基本ソフトウェアの11111の動作を示す。

【0023】

RTOS1121がS0において10 [ms] 周期タスク11111の起動処理を行う。その結果、10 [ms] タスクフレームワーク11111の処理が実行される。S1においてBIOS1122に外部入力処理を要求し、エアフローセンサ等の入力値を更新するセンサの最新値を取得する。S2において診断制御11112を実行する。その後いくつかの制御を実行し、S3において燃料補正制御11113を実行する。S4においてトルクベース制御11114を実行する。S5においてBIOS1122に外部への信号の出力処理を要求し、処理を終了する。アプリケーションソフトウェアを図5、図6に示すような構成にすることによって、外部からの入力処理S1と、異なる目的の制御処理S2、S3、S4、外部への出力処理S5を同じタスクの中で処理する事が可能になる。

【0024】

図7に、本発明が実施されたトルクベース制御アプリケーションソフトウェアの基本構造を示す。1Dは目標トルクの、1Eは変数Aの、1Fはスロットル開度の制御変数の更新ロジックを記述した制御ソフト部品である。2D、…、2E、2Fはそれぞれ2D、…、2E、2Fの制御ソフト部品に対応したインタフェースソフトである。3Aはトルクベース制御における変数の更新順序すなわち制御ソフト部品の実行順序を記述したトルクベース制御基本ソフトウェアである。4はBIOS1122から得られる外部入力データであって、例えばハードウェアの持つセンサで計測したり他のアプリケーションおよびコントローラと通信す

ることによって、前記外部情報を得る。5はBIOS 1 1 2 2によって出力される外部への情報の出力であって、ハードウェアの持つアクチュエータを駆動したり他のアプリケーションおよびコントローラと通信することによって、前記外部出力を実行する。

【0 0 2 5】

図7に示す構成にすることによって、例えば目標トルクを演算するソフト部品1Dが変更された場合にはインタフェースソフト2Dのみを変更すればよく、インタフェースソフト2E, 2Fは変更せずに用いる事ができる。すなわち図7の構成とすることによって、インタフェースソフトを有する組込みソフトウェアの再利用性が向上するという効果がある。

【0 0 2 6】

図8に、図7に示した制御ソフト部品1Fの一例を示す。ここで、C11はスロットル開度（以下TV0）を更新するソフト部品1をC言語で記述した場合のヘッダファイル(TV0_Calculate.h)である。C12は同様にTV0を更新するソフト部品1をC言語で記述した場合のソースファイル(TV0_Calculate.c)である。

【0 0 2 7】

C11は、TV0を更新する関数のプロトタイプ宣言を行っている。すなわち、関数の外部宣言であること(extern)と、関数の戻り値の型(void)と、関数名(TV0_Calculate)と、関数の入力値の型と変数名(unsigned short TargetTorque, ..., unsigned short Variable_A)と、関数が更新する変数つまりソフト部品の出力値の型と変数名のポインタ(unsigned short TV0)を宣言している。なお、関数の引数が入力変数か出力変数なのかということは、変数名の先頭にポインタ引数であることを示す識別子(*)がついているか否かで判断している。

【0 0 2 8】

なお前記出力変数は本発明における組込みコントローラにおいて、制御演算手段によって計算される制御データに対応する。

【0 0 2 9】

C 1 2 は、T V O を更新する関数を記述している。すなわち、関数の戻り値の型 (void) と、関数名 (TV0_CALCULATE) と、関数の引数である入力値の型と変数名 (unsigned short TargetTorque, ..., unsigned short Variable_A) と、関数が更新する変数つまりソフト部品の出力値の型と変数名のポインタ (unsigned short TV0) とを定義して、T V O の更新方法 ($*TV0 = TargetTorque * K_t + Variable_A * K_a$) を記述している。なお、TargetTorque および Variable_A は変数であり、 K_t および K_a は定数である。

【0030】

図 9 に、図 7 に示すインタフェースソフト 2 F の一例として C ソースコード C 2 1, C 2 2, C 2 3 を示す。これは、図 8 の C 1 1 および C 1 2 で説明した制御ソフト部品に対応する。C 2 1 は変数宣言を行う C ソースファイル (TV0.c) で、C 2 2 は変数参照命令を定義する C ヘッドファイル (TV0.h) であり、C 2 3 は変数更新命令を定義する C ヘッドファイル (TV0_Update.h) である。

【0031】

C 2 1 では、対応するソフト部品 C 1 1, C 1 2 の出力変数である T V O の変数宣言 (unsigned short TV0) を行っている。

【0032】

C 2 2 では、他のインタフェースソフトが C 2 1 で宣言した変数 T V O を参照するために、変数の参照する命令の定義 (#define TV0_Get() TV0) と、T V O の外部変数宣言として変数の型と名前の宣言 (extern unsigned short TV0) を行っている。C 2 3 では、制御基本ソフトウェア 3 が変数 T V O の更新を要求する際に実行する命令の定義を行っている。すなわち、入力となる変数を参照するための命令が記述された C ヘッドファイルを読み込み (#include "TV0.h" ~ #include "variable_A")、変数を更新するための命令名を定義し (#define TV0_Update() ¥)、実際に呼び出す制御ソフト部品 C 1 2 の関数名を記述し (TV0_CALCULATE ...)、T V O を更新する関数の入力となる制御変数を参照するための命令を呼び出し (TargetTorque_Get() ... variableA_Get())、T V O を更新する関数の出力として変数 T V O のアドレスを指定し (&TV0) している。

【0033】

図9において、インタフェースソフトC21～C23に対応するソフト部品C11、C12によって変数を更新するために呼び出される命令はTV0_Update()のみであるところに特徴がある。すなわち変数の更新を要求(TV0_Update())すると、インタフェースソフトが変数を更新する関数の呼び出しに必要な入力値を収集し(TargetTorque_Get()…variableA_Get())、出力となる変数を指定し(&TV0)、その後に変数を更新する関数を呼び出す(TV0_Calculate())。

【0034】

図9の構成とすることで、たとえば制御基本ソフトウェアから変数の更新を要求する場合、制御ソフト部品に記述された関数の入力値の数や型および変数名を全く意識する必要無く、共通の命令形式で変数の更新が行えるようになる。また、制御ソフト部品から入出力変数の宣言や更新等の操作を分離することが可能になり、制御ソフト部品の独立性が向上する。

【0035】

図10に制御基本ソフトウェアの一例として、エンジン出力を制御するトルクベース制御の制御基本ソフトウェア（以下トルクベース制御フレームワーク）C31を示す。

【0036】

C31では、図7のトルクベース制御基本制御部3A内で実行する制御ソフト部品をC311で定義し、対応するインタフェースソフトが定義する情報を読み込んでいる。また、制御基本ソフトウェア3内部で同時性が要求される変数の定義をC312において実行する。さらに、制御基本ソフトウェアで実行する入力処理・出力処理・OSサービス・ソフト部品の実行順序をC313において定義する。

【0037】

図11に、トルクベース制御基本制御部3Aの実行処理を示す。制御基本ソフトウェア3の実行順序は、図10のC313において定義されている。まずS31において割込み禁止処理(OSサービス)を呼び出し、外部信号に同期したタスク実行を禁止して入力値の同時性が保てるようにする。S32において外部入力データ4からエンジン回転数を読み込む。S33において外部入力データ4から

アクセル開度を読み込む。S 3 4 において割込み禁止解除処理（O S サービス）を呼び出す。S 3 5 において、目標トルクを示す変数の更新要求を行う（ソフト部品実行）。S 3 6 において、目標スロットル開度を示す変数の更新要求を行う（ソフト部品実行）。S 3 7 において、目標スロットル開度を外部の電子制御スロットルへ指示し（外部出力処理）、処理を終了する。

【0038】

制御基本ソフトウェアを図 1 0, 図 1 1 に示すような構成にすることによって、以下の利点がある。組込みソフトウェア中で変数の計算を行う部分すなわち図 7 の制御ソフト部品 1 D, …, 1 E, 1 F は、制御ロジックのデバッグや時定数など制御特性の調整に対応するために、変更が頻繁に行われる部分である。その一方で、変数の更新順序などを定めた部分すなわち図 1 0, 図 1 1 に示す制御基本ソフトウェアの部分のように、比較的変更が少ない部分も存在する。制御基本ソフトウェアを図 1 0, 図 1 1 に示すような構成にすることによって、変更が多い部分と変更が少ない部分とを分離する事ができる。そのため、制御基本ソフトウェア部分の再利用性が向上するという利点がある。

【0039】

図 1 2 は B I O S インタフェースソフトの構成の一例を示す。1 1 3 2 1 はエアフローセンサ入力に対応する B I O S インタフェースソフト、1 1 3 2 2 はクランク角度入力値に対応する B I O S インタフェースソフト、1 1 3 2 3 は燃料噴射量に対応する B I O S インタフェースソフト、1 1 3 2 4 は目標スロットル開度に対応する B I O S インタフェースソフトである。なお、B I O S インタフェースソフトの構成は、図 9 に示したインタフェースソフトの構成と同様である。B I O S インタフェースソフトを図 1 2 に示した構成にすることによって、たとえば制御基本ソフトウェアから外部入力値の更新を要求する場合、B I O S に記述された関数の入力値の数や型および変数名を全く意識する必要無く、共通の命令形式で外部入力値の更新が行えるようになる。外部出力に関しても同様である。また、B I O S から入出力変数の宣言や更新等の操作を分離することが可能になり、B I O S の独立性が向上する。

【0040】

図 13 に、組み込みソフトウェアの開発プロセスを示す。図 13 中におけるツールとは、プログラム、ソフトウェアおよび装置を含むものに対応するものである。制御ロジック設計ツール 61 によって制御モデル 62 を構築する。制御モデル 62 をもとに組み込みソフトウェア開発ツール 63 によって組み込みソフトウェアソースコード 64 を作成する。さらにソースコードコンパイルツール 65 を用いて組み込みソフトウェアのバイナリファイル 66 を作成し、ソフトウェア書込みツール 67 を用いて制御ユニット（ECU）10 に組み込みソフトウェアを書込む。

【0041】

以下、図 14，図 15，図 16 にソフトウェア開発ツール 63 の例を示す。

【0042】

図 14 に組み込みソフトウェア開発ツール 63 の一例として組み込みソフトウェア開発ツール 63A を示す。制御モデル 62 からソフト部品生成ツール 631A によってソフト部品 1 を生成する。インタフェースソフト生成ツール 632A は生成されたソフト部品 1 を参照して対応するインタフェースソフト 2 を生成する。

【0043】

図 15 に組み込みソフトウェア開発ツール 63 の一例として組み込みソフトウェア開発ツール 63B を示す。制御モデル 62 からソフト部品生成ツール 631B によってソフト部品 1 を生成する。インタフェースソフト生成ツール 632B はソフト部品生成ツール 631B からインタフェースソフト生成に必要な情報を受け取り、ソフト部品 1 に対応するインタフェースソフト 2 を生成する。

【0044】

図 16 に組み込みソフトウェア開発ツール 63 の一例として組み込みソフトウェア開発ツール 63C を示す。制御モデル 62 からソフト部品生成ツール 631A によってソフト部品 1 を生成する。インタフェースソフト生成ツール 632A は制御モデル 62 を参照して対応するインタフェースソフト 2 を生成する。

【0045】

図 14，図 15，図 16 に示す組み込みソフトウェア開発ツールの構成をとることによって、制御モデルから組み込みソフトウェアのためのソフト部品およびインタフェースソフトウェアを自動で生成することが可能になるため、人手でコード

イングする必要がなくなり、さらにコーディングミスが減少するためデバッグ作業を軽減することが可能になり、ソフトウェアの開発効率が向上する。

【0046】

続いて、図17、図18、図19においてインタフェースソフトの自動生成の具体的な手順を示す。

【0047】

図17に、インタフェースソフト生成手段632の処理を示す。S41において制御ソフト部品の情報を抽出し、S42においてインタフェースソフトを生成する。S41の詳細な内容は図18に、S42の詳細な内容は図19に示す。

【0048】

図18に、図8に示した制御ソフト部品を例に制御ソフト部品情報抽出処理S41の詳細を示す。まずS411においてファイル名を抽出する。すなわち、図8に示したC11およびC12のファイル名「TV0_Calculate.c」, 「TV0_Calculate.h」を抽出する。次に、S412において関数名を抽出する。すなわち図8に示したC11に記載された関数名「TV0_Calculate」を抽出する。S413において入出力変数の数を抽出する。すなわち図8に示したC12の関数「TV0_Calculate」の入出力変数をカウントする。S414において、S413においてカウントした変数の数に基づいて入出力変数名を抽出する。すなわち、図8に示したC12における「TargetTorque」や「variable_A」を入力変数として抽出し、「TV0」を出力変数として抽出する。なお、出力変数は変数名先頭にポインタ引数であることを示す「&」がついていることにより識別する。S415において入出力変数の型を抽出する。すなわち、図8に示したC12における「TargetTorque」の変数型「unsigned short」などである。

【0049】

前記入力変数は本発明における組込みコントローラ開発ツールにおいて、参照データに対応する。前記出力変数は本発明における組込みコントローラ開発ツールにおいて、制御データに対応する。

【0050】

図15に、インタフェースソフト生成処理S42の詳細を示す。図9に示した

インタフェースソフトの生成を例に説明する。S 4 2 1において、S 4 1 1で抽出したソフト部品のファイル名から、インタフェースソフトのファイルを生成する。すなわち、「TV0_Calculate.c」,「TV0_Calculate.h」から、区切り文字として組込みコントローラ内で意義する「_」以降の文字を消すことにより「T V O」というソフト部品名を得る。その上でソフト部品に対応するインタフェースソフトとして、「抽出したインタフェースソフトの名前」+「.c」というファイルと、「抽出したインタフェースソフトの名前」+「.h」というファイルと、「抽出したインタフェースソフトの名前」+「_Update.h」というファイルを生成する。すなわち、図9に示した「TV0.c」「TV0.h」「TV0_Update.h」である。次にS 4 2 2において、S 4 1 4で抽出したソフト部品の出力変数の名前とS 4 1 5で抽出したソフト部品の出力変数の型情報を持つソフト部品の出力となる変数の宣言部を「抽出したインタフェースソフトの名前」+「.c」というファイル内に生成する。すなわち、抽出した出力変数の型で、抽出した出力変数の変数名の変数をメモリ上に割り当てるための命令をソースファイルに出力する。図9に示したC 2 1における「unsigned short TV0;」である。この場合、「unsigned Short」の部分が出力変数の型で、「T V O」の部分が出力変数の変数名である。S 4 1 4で抽出したソフト部品の出力変数の名前をもとに、S 4 2 3においてソフト部品の出力変数の参照命令を生成する。すなわち、図9に示したC 2 2における「#define TV0_Get() TV0」である。これは、「出力変数の変数名」+「_Get()」という名前のマクロとして定義され、その実体は出力変数名になる。そのため、ソースコードにマクロを記述することによりプリプロセッサを用いてソースコードを変換すると、ソースコード内で参照用マクロは参照先の変数に置き換えられる。S 4 2 4において、S 4 1 4で抽出したソフト部品の入出力変数の名前とS 4 1 5で抽出したソフト部品の入力変数の型情報をもとに、出力変数の更新命令を生成する。すなわち、図9に示したC 2 3である。「出力変数名」+「_Update()」というマクロを出力変数の更新命令として定義する。そのマクロの実体はS 4 1 2で抽出した出力変数の演算関数「TV0_Calculate」に割り当てられ、関数の引数にはS 4 1 4で抽出した入力変数「TargetTorque」や「variable_A」の参照命令「TargetTorque_Get()」や「variableA_Get()」と出

力変数「TV0」のアドレスを示すポインタ「&TV0」を割り当てる。

【0051】

図17、図18、図19に示した構成を取ることによって、インタフェースソフト2を、制御ソフト部品1や制御モデル62から自動で生成することが可能になるため、人手でコーディングする必要が無くなり、さらにコーディングミスが減少するためデバッグ作業を軽減することが可能になり、ソフトウェアの開発効率が向上する。

【0052】

図20に、組込みソフトウェアのネットワーク配信の概要図を示す。図20ではソフト部品配信の例を示す。ネットワーク71に、ソフト部品サーバ73が接続されている。ソフト部品サーバ73はクライアントPC72をネットワーク71を介して接続され、クライアントPC72からの要求に応じてソフト部品1を送信する。クライアントPC72は受信したソフト部品1を電子制御ユニット（ECU）10Bに書込む事ができる。また、ECU10A自身にネットワーク接続機能を持たせることにより、自分自身でネットワーク71を経由してソフト部品サーバ73に接続し、ソフト部品を更新することができる。

【0053】

本実施例で示した組込みソフトウェアをネットワーク配信することによって、以下の利点がある。本発明の組込みソフトウェアでは、制御ソフト部品1が変更された場合は対応するインタフェースソフト2のみを変更すれば良く、制御基本ソフトウェアやタスク処理基本ソフトウェアなどの基本部分は変更する必要がない。そのため、サービス工場においてソフトウェアの変更をすることが容易になるという利点がある。また送信するデータ量も、全体を送信する場合に比べてソフト部品のみを送信する場合には少なくて済むという利点がある。

【0054】

図21に、組込みソフトウェア配信サービスの一例の概要を示す。クライアントPC72はネットワーク71を介して制御ソフト部品販売ホームページ74に接続する。制御ソフト部品販売ホームページ74にてダウンロードする制御ソフト部品1を選択したのち、料金決済システム75に接続し購入料金の決済を行う

。その後、制御ソフト部品サーバ73から制御ソフト部品1をダウンロードし、制御ユニット（ECU）10に書込む。

【0055】

図21で示した構成にすることによって、以下の利点がある。課金システムとホームページによる認証を行うことによって、サービス工場だけでなく一般のユーザーにとっても組込みソフトウェアの書込みが可能になる。すなわち、車両の走行モードや、自動変速機の変速パターン、さらには家電製品の組込みコントローラの組込みソフトウェアをネットワークで購入することによって、自分の好みの組込みコントローラを作る事ができるという利点がある。

【0056】

制御演算手段に組込みコントローラ内で用いる変数の宣言や入力処理および更新等の操作をさせてしまうと、制御演算手段に演算ロジック以外の機能が混在する。たとえば、演算結果を出力・保存する変数を組込みコントローラが持つメモリの特定のアドレスに割り当てる処理や、演算を行う際に他の制御演算手段の出力結果である変数を参照する処理や、制御基本手段が実行するための実行インタフェースの生成や、演算結果を出力する変数の参照処理などである。これは、演算ロジックの変更に伴い演算ロジック以外の機能変更及び検証が必要となることを意味し、組込みコントローラ用ソフトウェア開発に伴う工数が増大するという課題が生じる。本発明では、組込みコントローラにおいて、前記インタフェース手段が有する前記出力変数を他のインタフェース手段が参照する出力変数参照機能を、前記インタフェース手段に設けることにより、上記課題を解決する。本発明では、組込みコントローラにおいて、前記インタフェース手段に対応する前記制御演算手段が計算を行う際に用いる入力変数を、他のインタフェース手段もしくは自分自身の有する出力変数参照機能を用いるなどして収集する入力変数収集機能を前記インタフェース手段に設けることにより、上記課題を解決する。本発明では、組込みコントローラにおいて、前記制御基本処理手段などの命令で前記制御演算手段が計算を行う際に、前記インタフェース手段が前記入力変数収集機能により収集した前記入力変数を前記制御演算手段が参照できる状態にして、前記制御演算手段の計算結果が保存される出力変数を指定して、前記制御演算手段

の演算処理実行を命令する出力変数更新機能を前記インタフェース手段に設けることにより、上記課題を解決する。

【0057】

組込みコントローラ用ソフトウェアは、頻繁に変更される部分と半永久的に再利用される部分とに分類することが出来る。例えば、変数の演算ロジックのように変更されることが多い部分と、変数の更新順序のように変更されることが少ない部分とである。これが一体のプログラムとなっていると、組込みコントローラ用ソフトウェア内の変数の演算ロジックの変更であるにもかかわらず、変更する必要のない変数の更新順序の部分まで変更する必要があるという課題がある。本発明では、組込みコントローラにおいて、前記制御基本処理手段が、実行を命令する制御演算手段の種類と、前記制御演算手段と前記外部入力処理手段と前記外部出力処理手段と前記基本管理手段の実行順序とを定義し、前記制御基本処理手段が、前記外部入力処理手段と前記制御演算手段と前記外部出力処理手段と前記基本管理手段の実行を命令することにより、上記課題を解決する。

【0058】

一つの制御演算手段が複数の変数の演算を行う場合、ある変数の演算方法を変更するためには該当する制御演算手段を変更する必要がある。そのため、演算手段が変更されない変数に関しても、その演算を行う制御演算手段が変更されてしまうという課題がある。本発明では、組込みコントローラにおいて、前記制御演算手段の計算結果である、他の制御演算手段や外部出力手段が参照するための変数は、制御演算手段1つにつき1つとすることにより、上記課題を解決する。

【0059】

組込みコントローラのソフトウェアを変更する場合、ソフトウェアの一部のバージョンアップであるにもかかわらず、ソフトウェア全体をCD-ROMなどの記録媒体で配布する必要があった。そのため、ソフトウェアの配布に時間がかかるという課題や、CD等の媒体が必要になるために廃棄物が増大するという課題がある。本発明では、組込みコントローラ配信サービスにおいて、前記制御演算手段を保管および送信する制御演算手段サーバ手段と制御演算手段の販売を行う制御演算手段販売手段とをネットワークで接続することにより、上記課題を解決

する。

【0060】

車両制御に関しては、排気規制およびOBD規制などの法的規制、X-by-Wireおよびハイブリッドシステムに代表される制御システムのエレクトロニクス化を背景に、組込みコントローラ用制御ソフトウェアの高機能化・大型化が進んでいる。また、高機能化・大型化する制御ソフトウェアを効率良く設計・作成しなければならないという課題がある。本発明では、前記組込みコントローラが車両に搭載されてエンジンや変速機などの制御を行うことにより、上記課題を解決する。

【0061】

【発明の効果】

本発明によれば、組込みコントローラにおいて、制御演算手段毎に対応するインタフェース手段を設けることにより、組込みコントローラの再利用性が向上する。

【0062】

本発明によれば、組込みコントローラ開発ツールであって、前記制御演算手段がプログラミング言語のソースコードで関数として記述されて、センサ計測値や他の制御演算手段の計算結果などの計算に用いる変数は関数の引数で、計算結果である変数は関数の戻り値またはメモリ上のアドレスを指定した関数の引数であることと、前記インタフェース手段がプログラミング言語のソースコードで記述するとともに、前記制御演算手段のソースコードから特定の情報を抽出する制御演算手段解析手段と、制御演算手段解析手段の解析結果に基づいて制御演算手段に対応する前記インタフェース手段のソースコードを生成するインタフェース手段生成手段を設けることにより、組込みコントローラの生産性が向上する。

【図面の簡単な説明】

【図1】

本発明の図。

【図2】

実施例の車両用電子制御装置のシステム図。

【図 3】

実施例の分散組込みコントローラのシステム図。

【図 4】

組込みコントローラ用ソフトウェアの概要図。

【図 5】

アプリケーションソフトウェアの概要図。

【図 6】

タスク処理ソフトウェアのフローチャート。

【図 7】

トルクベース制御アプリケーションの構成図。

【図 8】

制御ソフト部品の C ソースコード。

【図 9】

インタフェースソフトの C ソースコード。

【図 1 0】

トルクベース制御フレームワークの定義情報。

【図 1 1】

トルクベース制御フレームワークのフローチャート。

【図 1 2】

分散制御システムにおける入出力処理部のシステム図。

【図 1 3】

組込みコントローラ用ソフトウェア開発プロセス図。

【図 1 4】

組込みソフトウェア開発手段 1。

【図 1 5】

組込みソフトウェア開発手段 2。

【図 1 6】

組込みソフトウェア開発手段 3。

【図 1 7】

インタフェースソフト自動生成フローチャート。

【図 18】

制御ソフト部品情報抽出フローチャート。

【図 19】

インタフェースソフト生成フローチャート。

【図 20】

組込みソフトウェアネットワーク配信図。

【図 21】

組込みソフトウェアネットワーク配信サービス図。

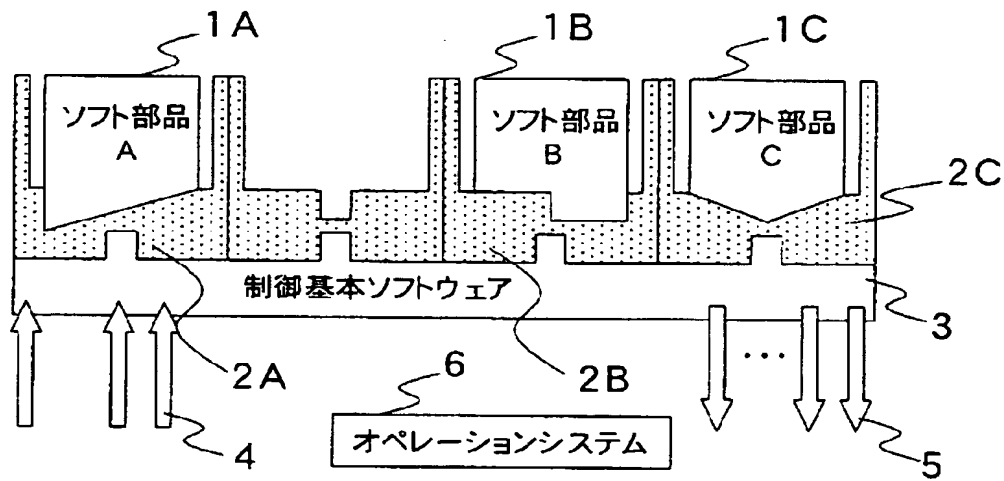
【符号の説明】

1…制御ソフト部品、1A…ソフト部品A、1B…ソフト部品B、1C…ソフト部品C、1D…目標トルクを示す変数を演算するソフト部品D、1E…変数Aを演算するソフト部品E、1F…スロットル開度を示す変数を演算するソフト部品F、2…インタフェースソフト、2A…ソフト部品Aに対応するインタフェースソフト、2B…ソフト部品Bに対応するインタフェースソフト、2C…ソフト部品Cに対応するインタフェースソフト、2D…ソフト部品Dに対応するインタフェースソフト、2E…ソフト部品Eに対応するインタフェースソフト、2F…ソフト部品Fに対応するインタフェースソフト、3…制御基本ソフトウェア、3A…トルクベース制御の基本制御部、4…外部入力データ、5…外部出力データ、6…組込みコントローラオペレーションシステム、7…車両、8…エンジン、9…自動変速機、A…アクセルペダル、10, 20…制御ユニット（ECU）。

【書類名】 図面

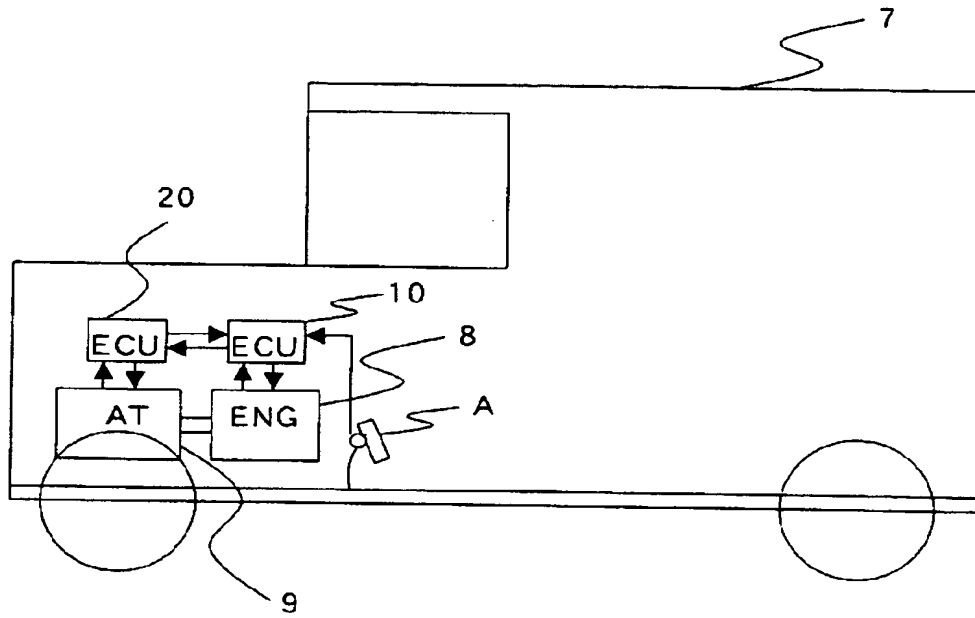
【図 1】

図 1



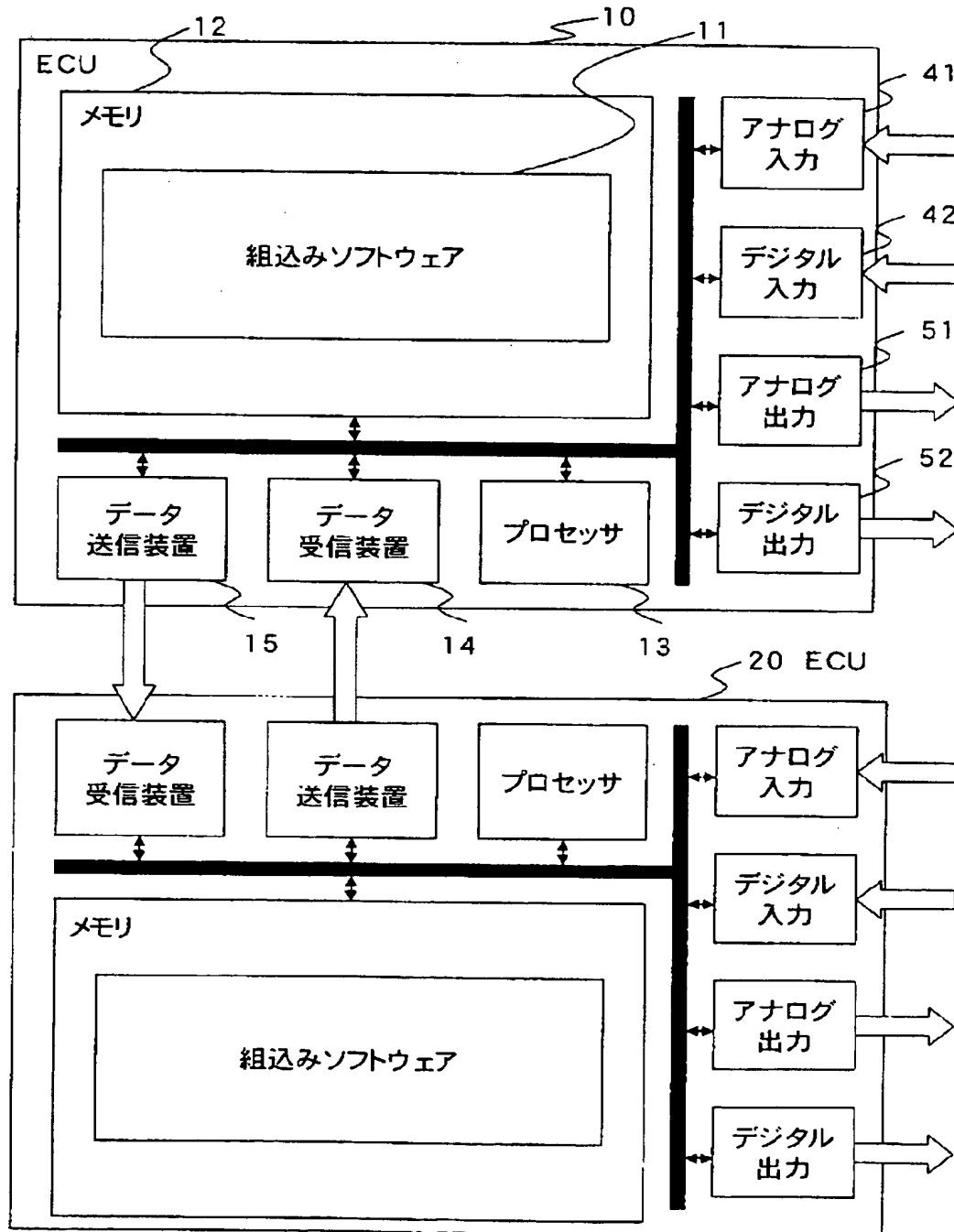
【図 2】

図 2



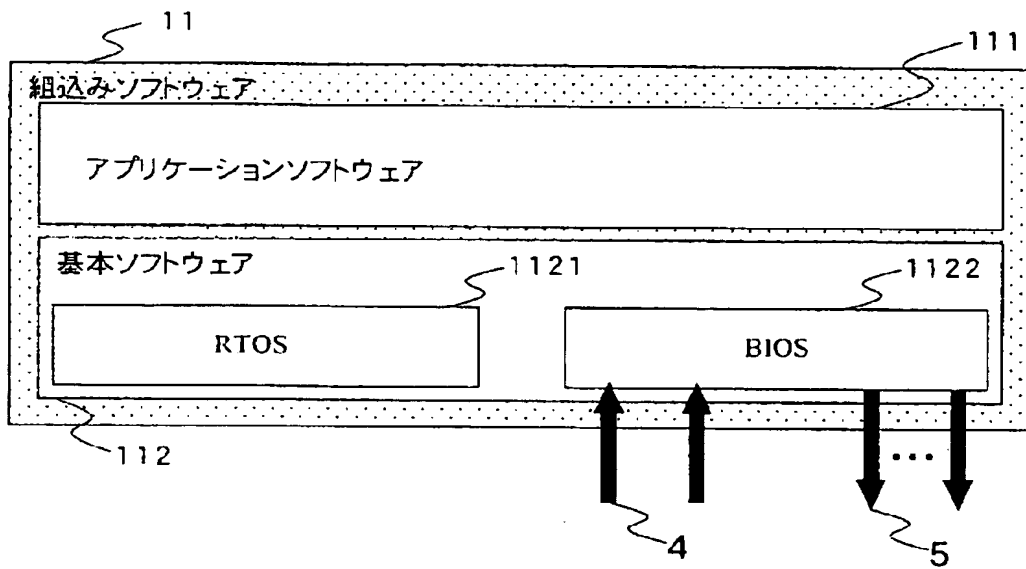
【図 3】

図 3



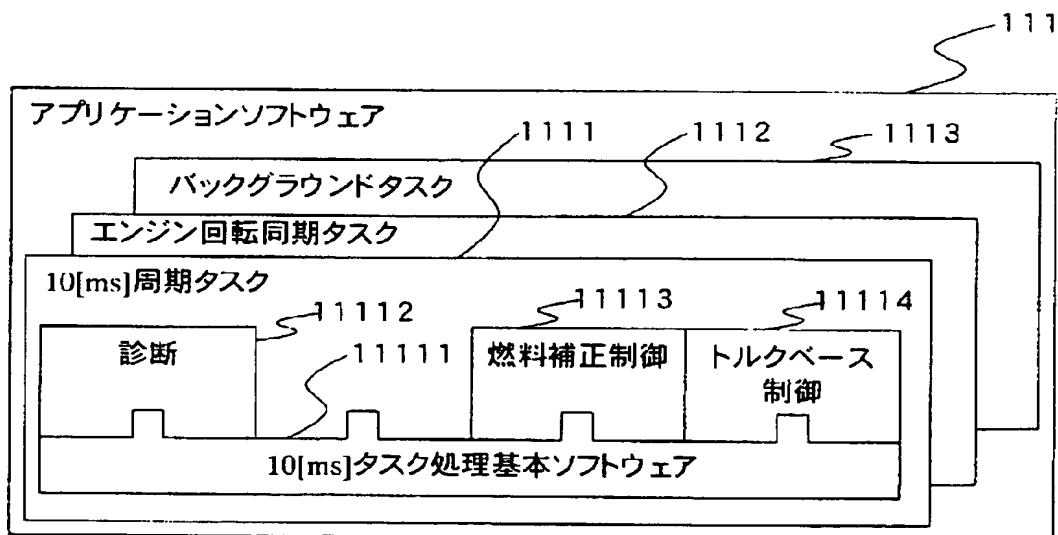
【図 4】

図 4



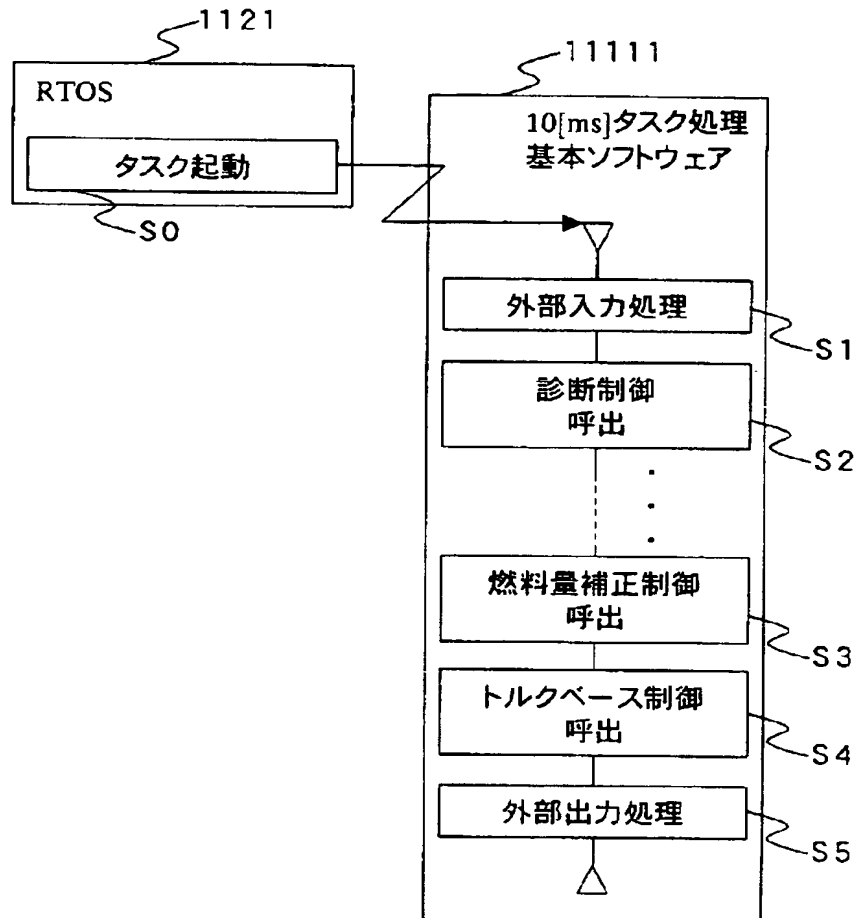
【図 5】

図 5



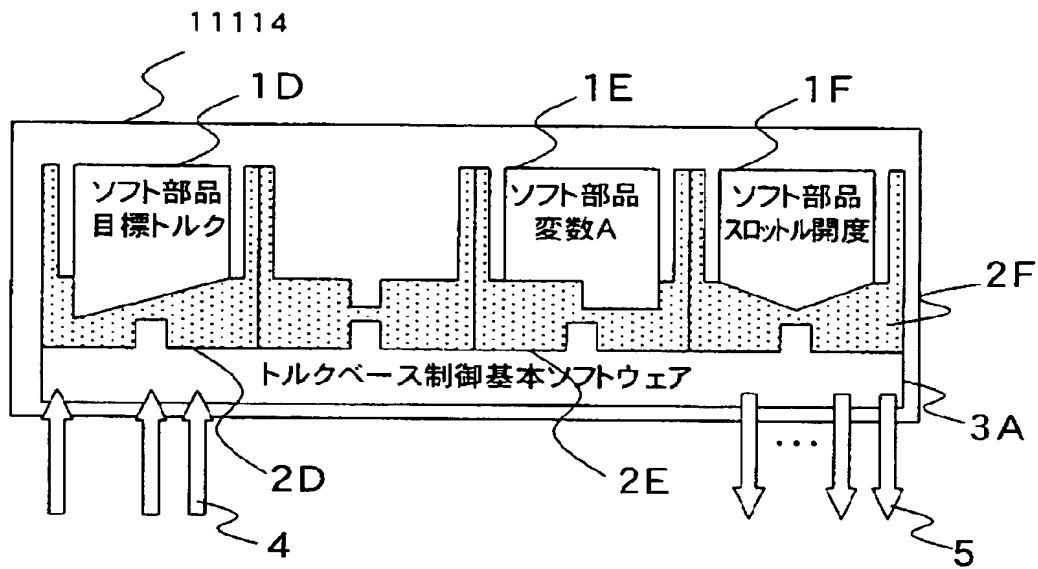
【図 6】

図 6



【図 7】

図 7



【図 8】

図 8

```
extern void TVO_Calculate(unsigned short TargetTorque,  
    ...,  
    unsigend short Variable_A,  
    unsigned short *TVO  
);  
  
/* EOF */
```

C11

```
void TVO_Calculate(    unsigned short TargetTorque,  
    .  
    .  
    .  
    unsigend short Variable_A,  
    unsigned short *TVO  
)  
{  
    .  
    .  
    .  
    *TVO = TargetTorque*Kt + Variable_A*Ka;  
}  
/* EOF */
```

C12

【図 9】

図 9

```
unsigned short TVO;
/* EOF */

#define TVO_Get() TVO
extern unsigned short TVO;
/* EOF */

#include "TVO.h"
#include "TargetTorque.h"
:
:
#include "variableA.h"
#define TVO_Update()
{
    TVO_Calculate(
        TargetTorque_Get(),
        :
        variableA_Get(),
        &TVO
    )
}
/* EOF */
```

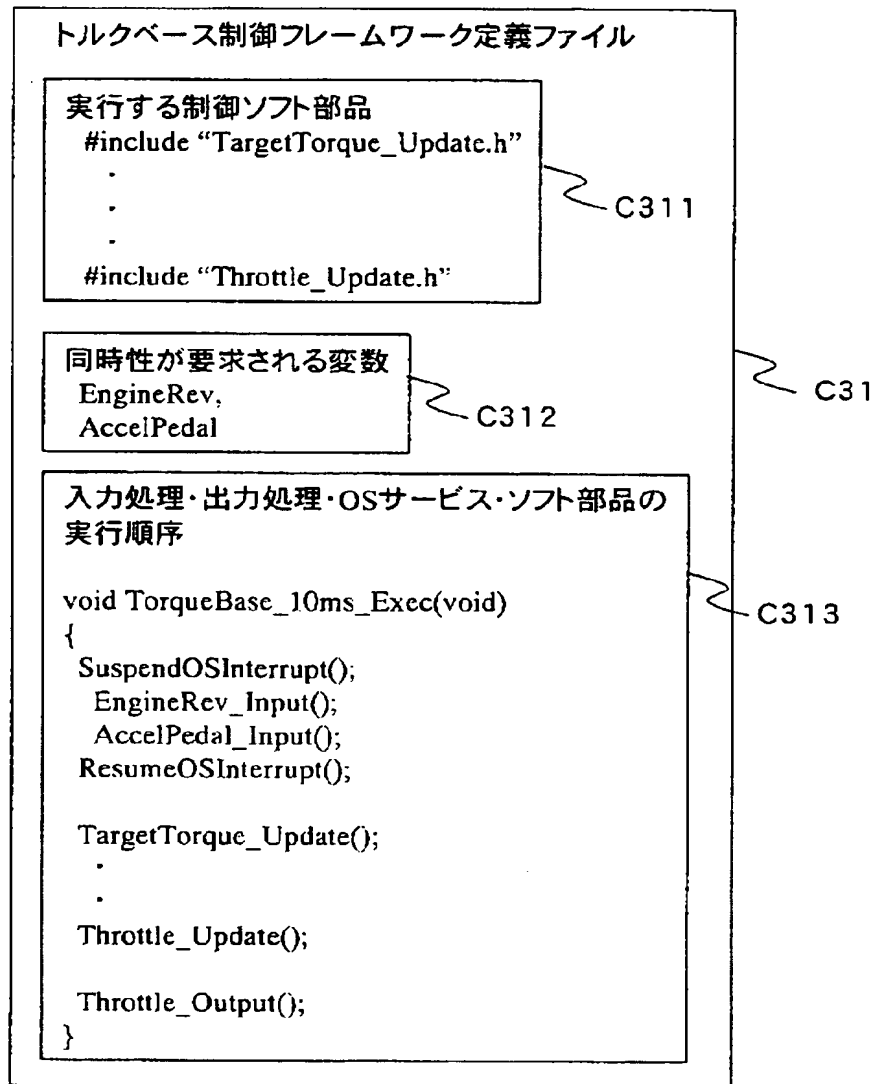
C21

C22

C23

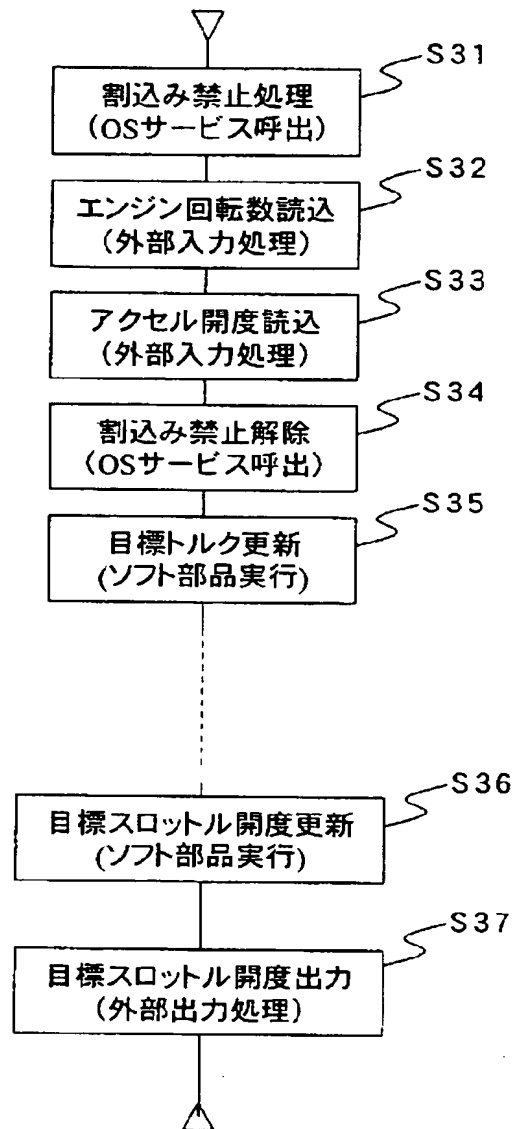
【図 10】

図 10



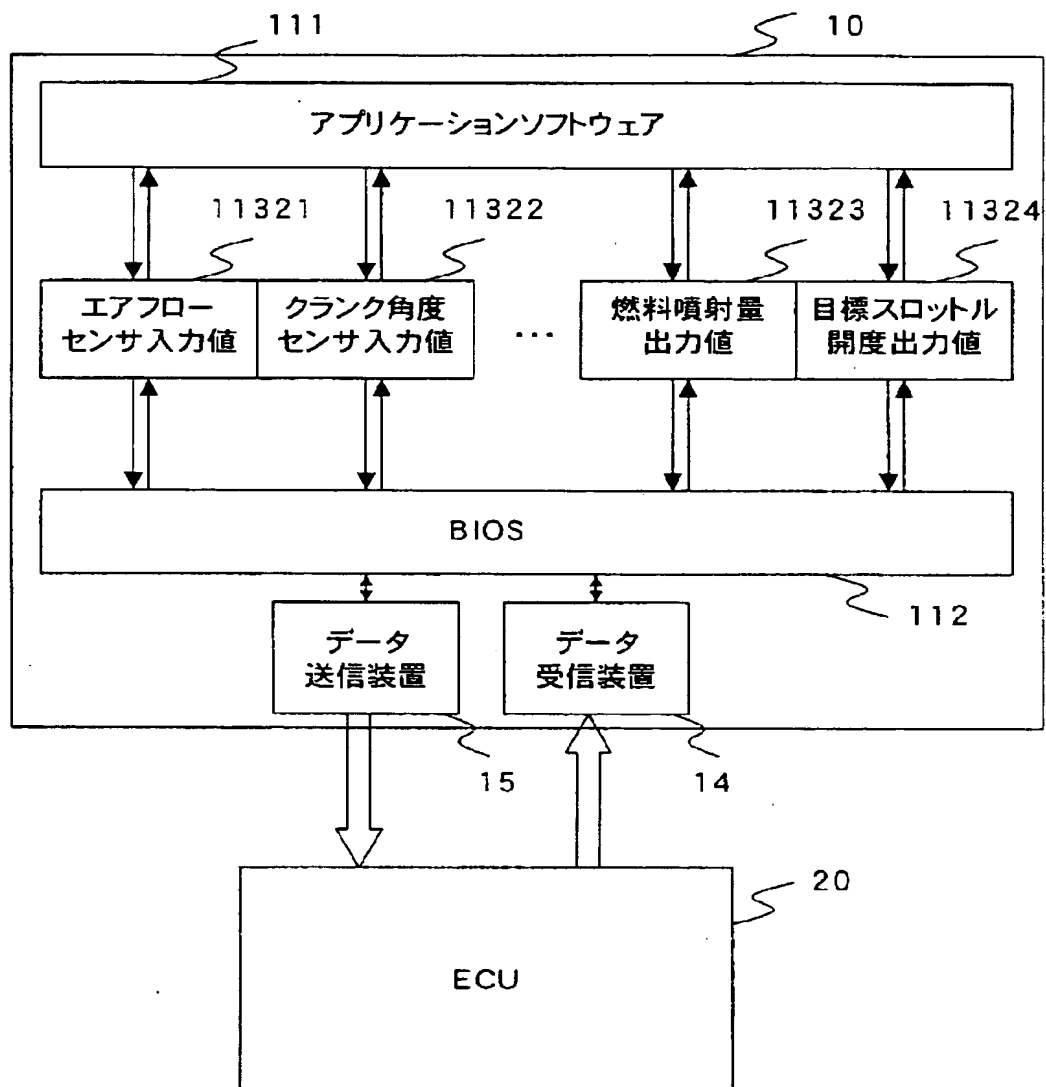
【図 11】

図 11



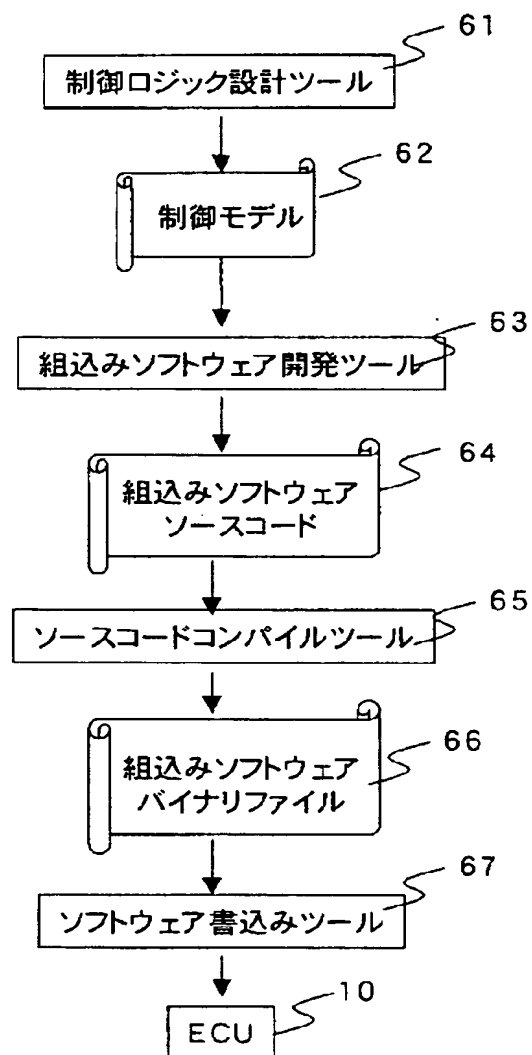
【図 12】

図 12



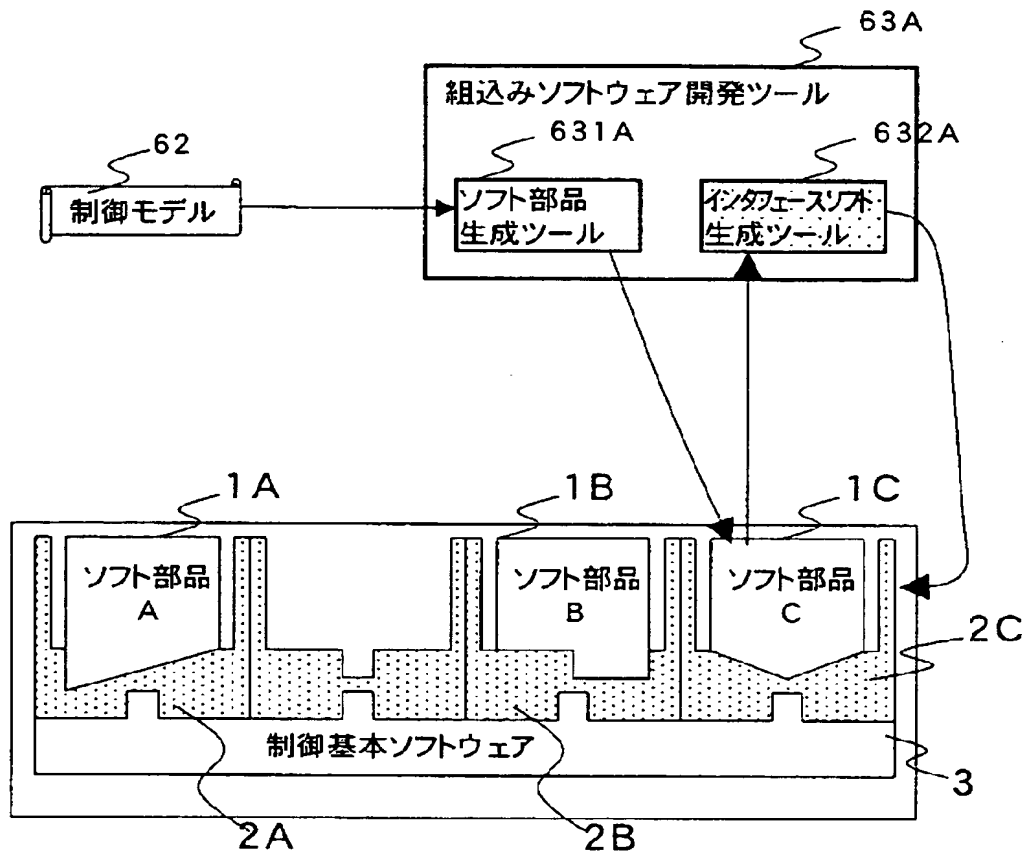
【図 13】

図 13



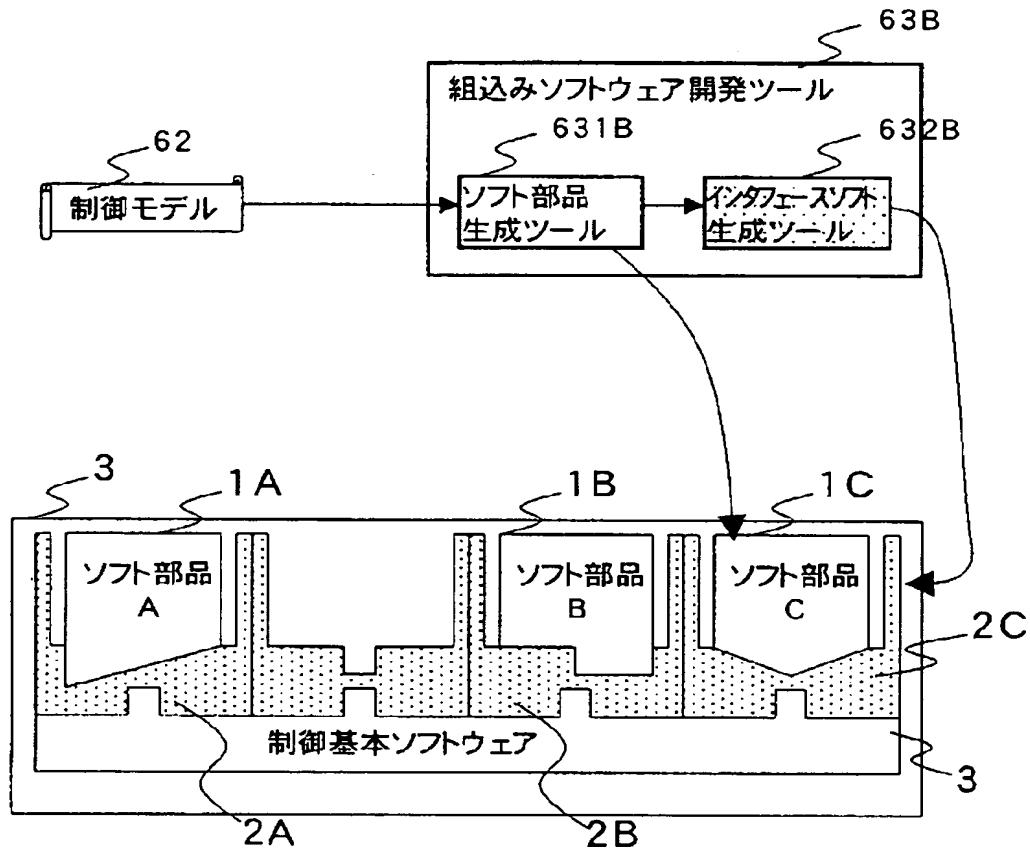
【図 14】

図 14



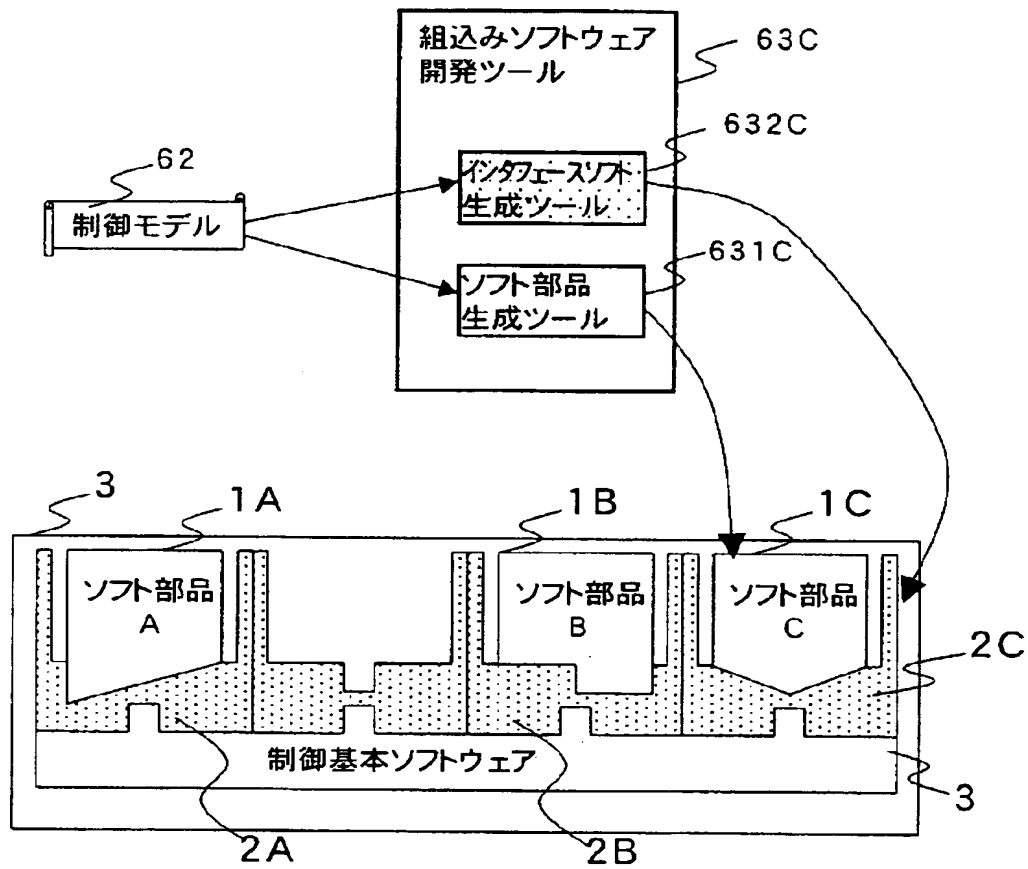
【図 15】

図 15



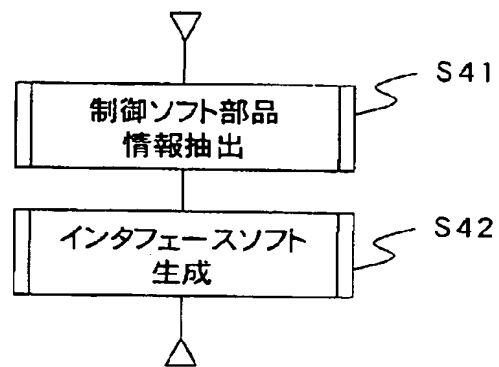
【図 16】

図 16



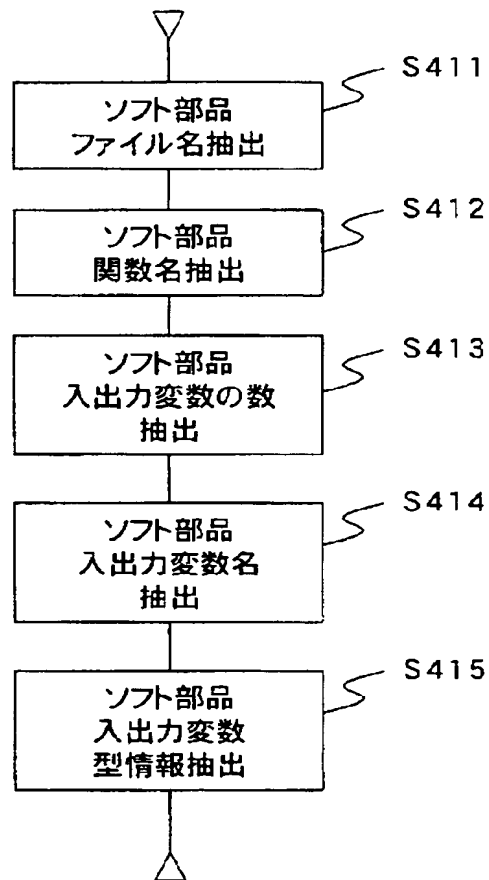
【図 17】

図 17



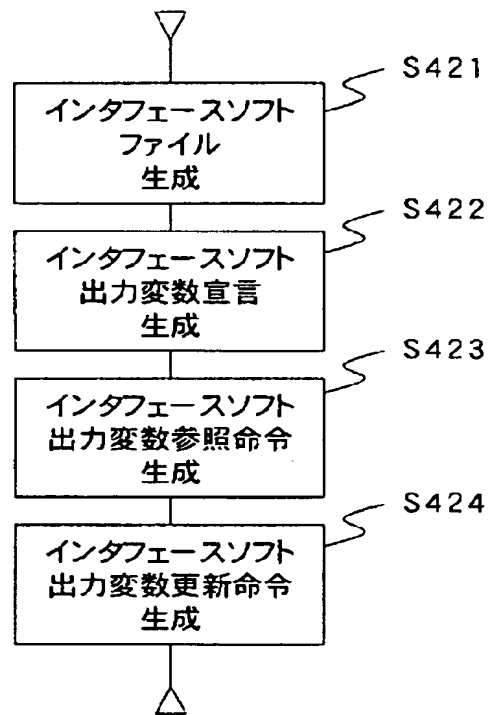
【図 18】

図 18



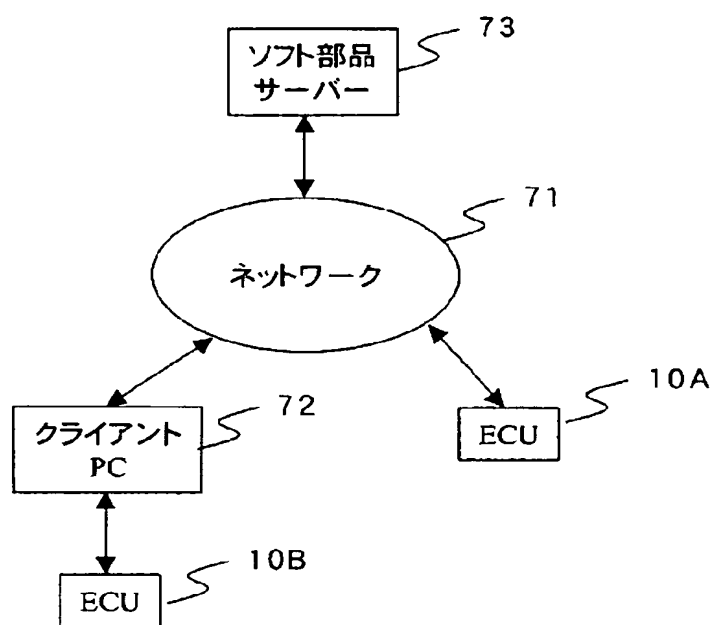
【図 19】

図 19



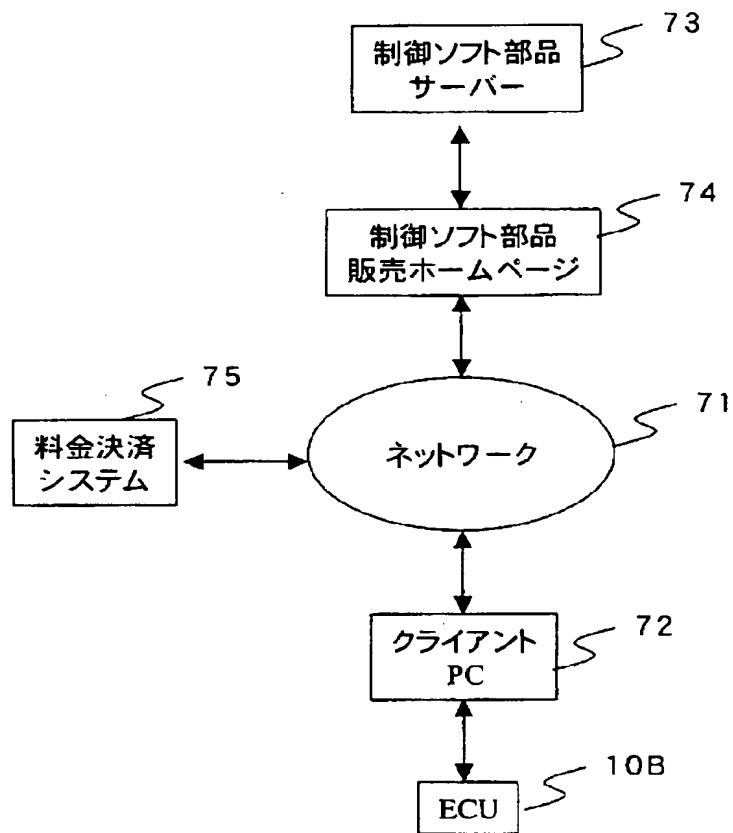
【図 20】

図 20



【図 21】

図 21



【書類名】 要約書

【要約】

【課題】

複数の制御演算手段と入出力処理手段と制御演算手段間および入出力処理手段などとのデータのやり取りを行うインタフェースソフトを有する組込みコントローラにおいて、制御演算手段の一部が変更された場合でもインタフェースソフトの再利用を可能にすることを目的とする。

【解決手段】

制御基本処理手段から制御演算手段への実行命令や前記制御演算手段間の変数の受け渡しなどを管理するインタフェース手段を制御演算手段ごとに設ける。

【効果】

一部の制御変数の演算ロジックを変更した場合にはインタフェースソフト全体を変更する必要は無く、対応するインタフェースソフトのみの変更で対応することが可能になり、組込みソフトウェアの再利用性が向上するという効果がある。

【選択図】 図 1

認定・付加情報

特許出願の番号	特願 2 0 0 2 - 3 6 9 1 6 7
受付番号	5 0 2 0 1 9 3 1 4 4 4
書類名	特許願
担当官	第七担当上席 0 0 9 6
作成日	平成 1 4 年 1 2 月 2 4 日

< 認定情報・付加情報 >

【提出日】 平成14年12月20日

次頁無



特願 2 0 0 2 - 3 6 9 1 6 7

出 願 人 履 歴 情 報

識別番号

[0 0 0 0 0 5 1 0 8]

1. 変更年月日

1 9 9 0 年 8 月 3 1 日

[変更理由]

新規登録

住 所

東京都千代田区神田駿河台 4 丁目 6 番地

氏 名

株式会社日立製作所